**APPLICAZIONI WEB: LINGUAGGI E ARCHITETTURE**

**a.a. 2017/2018**

**Progetto di laboratorio**

Alessandro Salogni, Riccardo Perotti, Simone Coppeta

20013907, 20013787, 20016455 @studenti.uniupo.it

1. **Istruzioni iniziali**

Collegarsi alla repository GitHub: <https://github.com/AlessandroSalogni/layw-appwebvc-2018> e scaricare il progetto. Una volta scaricato, aprirlo con Visual Studio 2017 (v. 15.8.2 o superiore), con .NET core 2.1.

Se ci si vuole loggare nell’applicazione come medico, è sufficiente sfruttare l’autenticazione di Google oppure quella di Facebook. Se invece ci si vuole loggare come utente amministratore, si deve premere il link apposito ed inserire mail e password (comunicati successivamente). Al primo login da medico, naturalmente, non ci saranno pazienti associati; pertanto ci si deve loggare come amministratore (account: [mantovani@mail.com](mailto:mantovani@mail.com) password: mantovani). Dopodiché sarà possibile collegare dei pazienti all’account medico precedentemente creato.

Non è necessario alcuno script SQL o altre impostazioni iniziali. I dati sorgenti sono estratti tramite API REST, eseguite su un server Heroku, ove è contenuto anche il Database.

1. **Scelte implementative e note particolari**

L’applicazione Web dialoga con lo strato di API REST definite sul server che rappresentano l’accesso al modello dei dati, rappresentato mediante componenti grafiche del framework Kendo. Le API REST sono state sviluppate in Java per il progetto di Reti 2.

Essendo che, lato server abbiamo implementato un MQTT publisher, lato client in questa applicazione Web abbiamo implementato un MQTT listener per permettere l’ascolto dei messaggi pubblicati e gestire così le varie notifiche da visualizzare al dottore.

Nella homepage abbiamo deciso di rappresentare, tramite grafici a torta, i pazienti che hanno superato specifici obiettivi per permettere al dottore di avere una visione generale ma allo stesso tempo diretta della situazione globale dei suoi pazienti; mentre per tutte le altre informazioni relative ad un paziente (come la dieta, l’allenamento ecc..), tramite apposito menù il dottore può visionarle singolarmente.

Due tipi di login, perché staccati, pro e contro.

Il sistema è pensato per una piccola utenza per ogni medico. Si suppone che i casi che segue un determinato medico siano sufficientemente pochi per consentirgli un’attenta analisi giornaliera. Si stimano 5-10 pazienti massimo per ogni medico.

1. **Descrizione del modello dei dati**

Vedi allegato per la descrizione delle API REST sviluppate (in Java poiché facevano parte del progetto di Reti 2)

**ALLEGARE DOCUMENTO**

Per la gestione degli utenti amministratore è presente un db locale all’applicazione Web contenente una sola tabella avente due campi: e-mail e password.

1. **Descrizione del modello ad oggetti**

UML